

# ガイスター AI の研究

川上 直人<sup>1,a)</sup> 橋本 剛<sup>1,b)</sup>

**概要:** 近年, 囲碁や将棋の AI が人間のトッププレイヤーに勝利するなど完全情報ゲームの研究は大きな成果を上げており, 次のターゲットとして不完全情報ゲームが注目されている. バックギャモン, ポーカー, 麻雀においてはトッププレイヤー相当の実力を持つ AI が研究されているが, ガイスター・軍人将棋など, チェスや将棋とルールが似ている不完全情報ボードゲームでは強い AI の研究があまり行われていない. 本稿では不完全情報ボードゲームの一つである「ガイスター」AI を題材とし, GPW 杯ガイスター AI トーナメント 2017 で優勝した AI のアルゴリズムを紹介する.

## Research of Geister AI

KAWAKAMI NAOTO<sup>1,a)</sup> TSUYOSHI HASHIMOTO<sup>1,b)</sup>

### 1. はじめに

コンピュータの性能・学習アルゴリズムの発達によって, 完全情報ゲームの研究は大きな成果を上げている. 2015 年には囲碁 AI 「AlphaGo[1]」が囲碁トッププレイヤーのイ・セドル氏に 4 勝 1 敗で勝利し, 世間からの注目を集めた. 完全情報ゲームのいくつかはコンピュータプレイヤーが人間トッププレイヤーの実力を上回ったため, 次のターゲットとして不完全情報ゲームが注目されている.

不完全情報ゲームでも, バックギャモン, ポーカー, 麻雀においてはトッププレイヤー相当の実力を持つ AI が研究されている. しかし, ガイスター・軍人将棋など, チェスや将棋とルールが似ている不完全情報ゲームでは, 強い AI の研究があまり行われていない. そこで, チェスや将棋とルールが似ている不完全情報ゲームを対象とし, 高い実力を持つ AI の研究をすることにした. 本稿では, 不完全情報ボードゲームの一つである「ガイスター」を対象とする. ガイスターを選んだ理由は, ルールが単純でプログラムの性能評価がしやすいと考えたためである. 本稿では, 2017 年 11 月に開催された「GPW 杯ガイスター AI トーナメン

ト 2017」で優勝した AI のアルゴリズムを紹介する.

2 章では, ガイスターのルールについて説明する. 3 章では, 先行研究について説明する. 4 章では, 本稿で紹介するアルゴリズムの方針を述べる. 5 章では, 作成した AI のアルゴリズムを紹介する. 6 章では, 作成した AI に用いた評価関数を紹介する. 7 章では, 実験方法について説明する. 8 章では, 実験結果について説明する. 9 章では, 第一回 GPW 杯ガイスター大会の結果を述べる. 10 章では, 考察をする. 11 章では, まとめをおこなう.

### 2. ガイスターのルール

ガイスターは相手の駒の色を観測できない状態で進行する 2 人対戦ボードゲームである. 駒の色はゲームの勝敗に関係する. 図 1 はガイスターの盤面である. 十字方向の隣接したマスに移動できる駒を互いに 8 個 (青 4 個, 赤 4 個) 持ち, 先手後手が交互に駒を動かしていく. 盤面は 6 \* 6 マスであり, 最初 8 個の駒を置くマスは決まっているが, どの色の駒をどのマスに置くかはプレイヤーが自由に決めることができる. 駒は基本的に盤外に脱出できないが, 相手陣の矢印マスにある青い駒だけは盤外に脱出させることができる. 勝利条件は 3 種類あり, それぞれ, 相手の青い駒を全て取る・自分の赤い駒を全て取らせる・自分の青い駒を 1 匹盤外へ脱出させる, のいずれかである. このゲームでは, 完全情報ゲームとは違い, 完全な読みはできないことが多い.

<sup>1</sup> 情報処理学会

IPSJ, Chiyoda, Tokyo 101-0062, Japan

<sup>†1</sup> 現在, National Institute of Technology Matsue College

<sup>a)</sup> s1711@matsue-ct.jp

<sup>b)</sup> hashimoto@matsue-ct.jp

### 3. 先行研究

先行研究には、ニューラルネットワークを用いた Q 学習 [2]、モンテカルロ法 (UCT) による探索 [3] がある。文献 [2] では、「相手の駒の色配置が未知な盤面 S」について指し手 A を選択した場合の勝率  $Q(S, A)$  をニューラルネットワークを用いた強化学習によって近似している。文献 [3] では、相手の駒の色配置を確率推定し、モンテカルロ法を用いて完全情報の探索をし、平均勝率の高い指し手を選択している。すなわち、下記の方法を用いている。

- 世界 (相手の駒の色配置) を確率推定 → プレイアウト回数の割り振りに使用。
- UCT によって、完全情報ゲームの探索。各世界における各指し手の勝率推定。
- 各自分の指し手について、勝率の平均 (全世界の勝率の和 / 世界の数) を計算し、それが最大の指し手を選択する。

文献 [2] ではランダム AI に対する勝率が 75% 程度であり、そこまで強くない。文献 [3] では「ゴールに最も近い青駒をゴールへ近づける戦略」に対して 95% 程度の勝率を上げており、思考時間も 1 秒程度と少なく、実用的である。そこで、文献 [3] をベースとすることにした。

### 4. 方針

ガイスターでは、「相手の駒の色配置」という不確定要素がある。しかし、初期配置が  $8C4 = 70$  通りしかない。そのため、実用的な時間で、全てのありえる盤面についてミニマックス探索などを適用することはそれほど難しくはない。そこで、本稿で紹介するアルゴリズムでは、完全情報ゲームの探索手法を用いることにした。

本稿で紹介アルゴリズムでは、文献 [3] と同様、完全情報盤面の探索結果を集計するが、探索方法 (と探索結果の集計方法) を変更する。探索方法を変更する理由は、モンテカルロ法よりもゲーム木探索の方が有効な場合があるのではないかと考えたためである。モンテカルロ法は囲碁や将棋など、探索空間が広い場合は有効である。しかし、ガイスターのように探索空間が狭いゲームでは、UCT よりも、ミニマックス法などを用いて基本的に全部調べたほうが、手軽に強いものが作れるのではないかと予想した。ミニマックス法はチェスや将棋など、完全情報二人ゲームで大きな成果を上げている。したがって、オープンソースで有名なチェスプログラム「StockFish」などを参考に、ガイスター AI にミニマックス探索を導入することで強いプログラムになるのではないかと期待できる。

### 5. アルゴリズム

- 本章では、ガイスター AI のアルゴリズムを 2 つ紹介する。
- 入力：相手の駒の色配置が未知な盤面

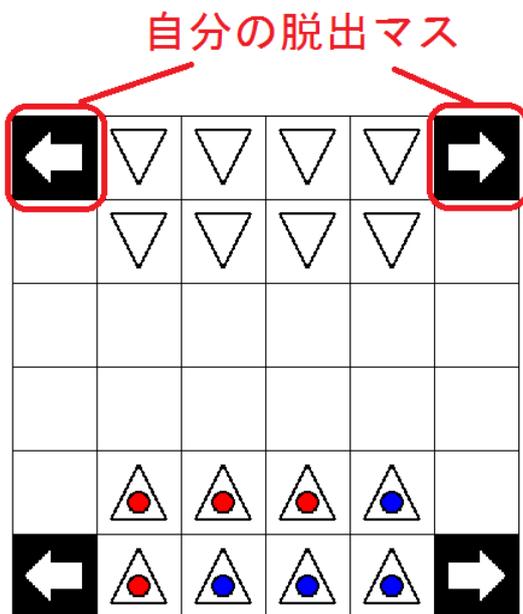


図 1 ガイスター

- 出力：指し手

という関数を設計する。

モンテカルロベースの先行研究では、各指し手の平均勝率を元に指し手を決めていた。それに対し、本稿では以下の2つの方法を考える。

- (1) 多数決. 各「相手の駒の色配置」においてミニマックス法によって調べられる範囲で最善な手を計算し、最も多く採用された手を選択する。
- (2) ミニマックス. 各指し手について最悪評価値を計算し、それが最も高かった手を選択する。最悪評価値は、各「相手の駒の色配置」において「指し手」を指した直後の盤面からミニマックス探索をしたときの評価値の最小を取ったものとする。評価値は手番プレイヤーの有利さを表すヒューリスティックな値とし、これが高いほど有利な盤面になるように設計する。

また、初期配置はどちらも「ランダム」とし、探索の深さは（開始盤面を深さ0として）4~5程度とした。

アルゴリズム1は「最善な指し手は、多くの想定盤面において最善であろう」という予想から自然に思いついたアルゴリズムである。アルゴリズム2は「無難な手を指す」という方針から自然に思いついたアルゴリズムである。

GPW杯ガイスターAIトーナメント2017で優勝したAIでは、アルゴリズム2を用いた。アルゴリズム2の具体例を図2に示す。この例では、指し手Hand2またはHand3をランダムで選択している。図2の各（盤面、指し手）に書かれた評価値は、説明のためデータラメに設定したものであり、次章で説明する評価関数とは関係ない。アルゴリズム1,2では完全情報盤面の探索において、どちらもミニマックス探索を用いている。ミニマックス探索の内部では「互いに駒の色が分かっている状態のガイスター」を考えている。

何回か実験をしたところ、アルゴリズム1は赤い駒4個を取って負ける・相手の青い駒の脱出を許して負けるケースが目立ち、そこまで強くないように思えた。一方で、アルゴリズム2ではこのような負け方はあまり存在せず、ランダムAIや「ゴールに最も近い青駒をゴールへ近づける戦略」および「その逆」と対戦すると、高確率で勝てたため、そこそこ強いように思えた。ここはあくまでも予想であり、どちらが強いかわかるかは、実際に勝率を取って見ないと判断できないが、ガイスターAI大会ではアルゴリズム2を採用した。

## 6. 評価関数

探索の葉ノードで用いられる評価値には「青駒の個数の差」を用いた。ただし、手番プレイヤーが勝った場合は「∞点」、負けた場合は「-∞点」とした。「青駒の個数」に注目したのは、「青い駒が残っているプレイヤーの方が、相手に取られて負けるリスクが減り、自分が脱出できる可能性が高くなるだろう」と予想したからである。「赤駒の個数」

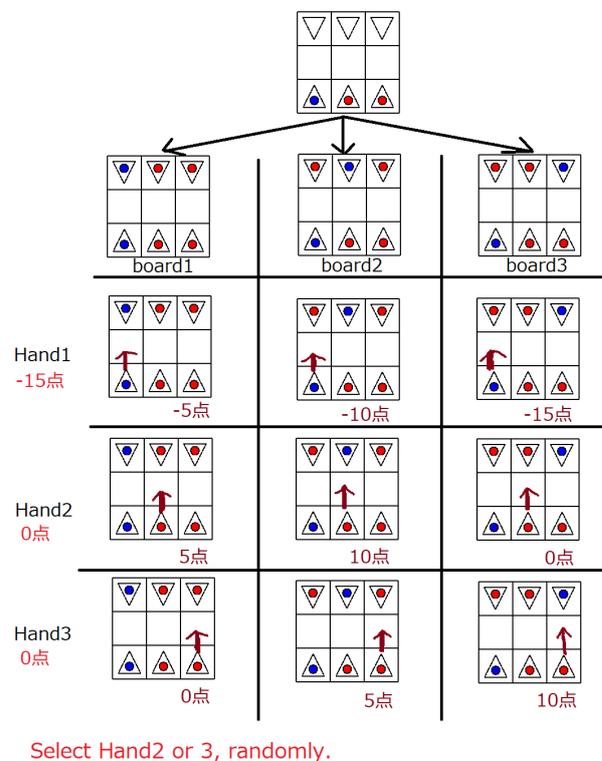


図2 アルゴリズム2の具体例

は多い方が良いか少ない方が良いか予想が立たなかったため、今回は評価に入れていない。この評価関数はそこまで強くないと考えられるが、実装が簡単であるため、まずはこの方針で実装することにした。

## 7. 実験方法

勝率の評価、実行時間の計測のために、上記のアルゴリズムを実装したプログラムを用いて実験をおこなった。以下の実験をおこなった。

- (1) 猪突猛進戦略, アルゴリズム 1, アルゴリズム 2 を手番入れ替えを考慮した総当たりで対戦させる (全 6 試合).  
本実験では、各組み合わせの対戦数を基本的に 50 試合とした。ただし、実験時間の都合上、アルゴリズム 1 とアルゴリズム 2 の対戦は各 5 試合とした。ここで、猪突猛進戦略とは、最もゴールに近い青駒をゴールへ近づける戦略である。猪突猛進戦略の詳細については、文献??に記載されている。

## 8. 実験結果

実験結果を図 3 に示す。図 3 では、「先手 vs 後手」の「勝ち数 vs 勝ち数」を記録した。EvalAI4 はアルゴリズム 1, MinMax はアルゴリズム 2, SimpleAI は猪突猛進戦略を実装した AI である。図 3 から分かる通り、猪突猛進戦略との対戦において、アルゴリズム 2 はアルゴリズム 1 よりも勝率が高かった。また、simpleAI が先手 EvalAI4 が後手の対局では、EvalAI4 が 49 勝 1 敗で圧勝した。対戦履歴を解析してみたところ、EvalAI4 が勝利したときの駒の動かし方は（初期配置によらず）同じになっていた。

一方で、アルゴリズム 1 は猪突猛進戦略と先手番で対戦したとき、勝率が 56%であった。また、アルゴリズム 1 とアルゴリズム 2 を対戦させたところ、8 つの試合では盤面が周期的に振動してしまい引き分けになってしまったが、残る 2 試合ではアルゴリズム 2 が勝利した。

## 9. 第一回 GPW 杯ガイスター大会の結果

GPW 杯ガイスター AI トーナメント 2017\*1 を開催し、11 月 10・11 日にガイスター AI のリーグ戦をおこなった。アルゴリズム 2 を実装したプログラム (なおっち MINMAX) を含めて 6 つのプログラムが参加し、各組み合わせについて先手番を入れ替えて 2 試合おこない、勝ち数の総数を競った。時間制限は 1 試合のトータル持ち時間 600 秒、時間切れ後は 1 手 10 秒読みとした。

結果を図 4, 5, 6 に示す。図 4 の各マスに書かれている ○ × は勝敗を表している。i 行 j 列目のマスの 1 文字目は i 行目のプレイヤーが先手で j 列目のプレイヤーが後手の場合の、i 行目のプレイヤーの勝敗を表しており、2 文字目は手番を逆にした場合の勝敗を表している。アルゴリズム 2 を実装した

```
simpleAI vs MinMax↓  
3 vs 47↓  
↓  
MinMax vs simpleAI↓  
45 vs 5↓  
↓  
EvalAI4 vs simpleAI↓  
28 vs 22↓  
↓  
simpleAI vs EvalAI4↓  
1 vs 49↓  
↓  
EvalAI4 vs MinMax↓  
0 vs 1↓  
(draw:4)↓  
↓  
MinMax vs EvalAI4↓  
1 vs 0↓  
(draw:4)↓
```

図 3 3 プログラムの総当たりの結果

\*1 GPW 杯ガイスター AI トーナメント 2017 :  
<http://www2.matsue-ct.ac.jp/home/hashimoto/geister/>

プログラム (なおっち MINMAX) は、同率 1 位のプログラムとのプレイオフの結果、優勝することができた。しかし、初心者人間プレイヤー 2 人と対戦したところ、連敗した。

	な	立木	旧立	PF	PI	H
なおっち MINMAX		OO	OO	OO	XO	OO
立木システム プレイヤー	XX		OO	XX	XX	XO
旧立木システム プレイヤー	XX	XX		XO	XX	XX
POMCP Fastest	XX	OO	XO		XX	XX
POMCP Ichi	OX	OO	OO	OO		OO
Haunter	XX	XO	OO	OO	XX	

図 4 リーグ表

	1 戦目	2 戦目
POMCP Ichi	先手×	後手×
なおっち MINMAX	後手○	先手○

図 5 プレイオフの結果

AI の名前	勝敗	順位
なおっち MINMAX	9-1	1
立木システム プレイヤー	3-7	4
旧立木システム プレイヤー	1-9	6
POMCP Fastest	3-7	4
POMCP Ichi	9-1	2
Haunter	5-5	3

図 6 勝敗と順位 (勝ち数の合計でソート)

## 10. 考察

実験では、猪突猛進戦略と対戦した場合、アルゴリズム 2 は手番によらず安定した強さを持っていた。勝率は文献 [3] に若干劣ったが、これは評価関数を強化するなどすればもう少し改善すると考えられる。アルゴリズム 1 とアルゴリズム 2 の対戦では引き分けが目立ったため、直接対決による優位性は確認できなかったが、図 3 の「EvalAI4 vs simpleAI」の対戦結果と「MinMax vs simpleAI」の対戦結果を比較すると、アルゴリズム 2 の方が多くの場面で安定して勝率を上げられると考えられる。実際に、ガイスター AI トーナメントで「アルゴリズム 2」を用いた結果、優勝することができたため、この方針は上手くいったのではないかと考えられる。

しかし、ガイスター AI トーナメントを観戦していた初心者人間プレイヤーとアルゴリズム 2 を実装したプログラムを対戦させたところ、プログラム側が 2 連敗してしまった。これは、アルゴリズム 2 では各手について最悪な盤面を想定しているため、読みの手数が少ないと、どうしても取らないといけない局面まで相手の駒を避けつづけてしまう傾向がある。よって、赤を 3 つ特攻されると弱いと予想される。実際、連敗したときの内容はいずれも、人間側が赤い駒を 3 つ相手側の隅マスまで移動させ、それを AI が取ってしまい、その後人間が AI の青い駒を 4 個取るというものであった。このように、少し対策すると初心者でもこの AI に勝つことは簡単にできてしまうと考えられる。

また、先ほど紹介した 2 つのアルゴリズムでは、初期配置はランダムで設定するが、指し手は与えられた盤面のみで確定してしまう。よって、この戦略を相手が実装してしまえば、指された手から駒の色配置を限定されてしまい、とても不利になると考えられる。また、学習もしていないため、本稿で紹介したアルゴリズムを知らなくても何回か対戦することで戦略を推定できてしまう可能性がある。今後は、確率的な選択または学習の導入も検討していきたい。

## 11. まとめ

チェスや将棋とルールが似ている不完全情報ゲーム「ガイスター」を対象に、対戦プログラムを作成した。ガイスター AI トーナメントでは優勝することができたため、本稿で提案したアルゴリズム 2 はそこそこ有効であると考えられる。しかし、初心者の人間プレイヤーには負けてしまったため、さらなる改善が必要である。

## 参考文献

- [1] David Silver, Aja Huang 他, Mastering the game of Go with deep neural networks and tree search, nature16961 (2015)
- [2] 佐藤 佑史, ガイスターにおける自己対戦による行動価値関数の学習, 組み合わせゲーム・パズルプロジェクト第 11 回研究集会 (2016)
- [3] 三塩 武徳, 小谷 善行, ゲームの不完全情報推定アルゴリズム UPP とそのガイスターへの応用, 情報処理学会研究報告, Vol.2014-GI-31, No.4 (2014)